



Beginning Programming:

Learning Python

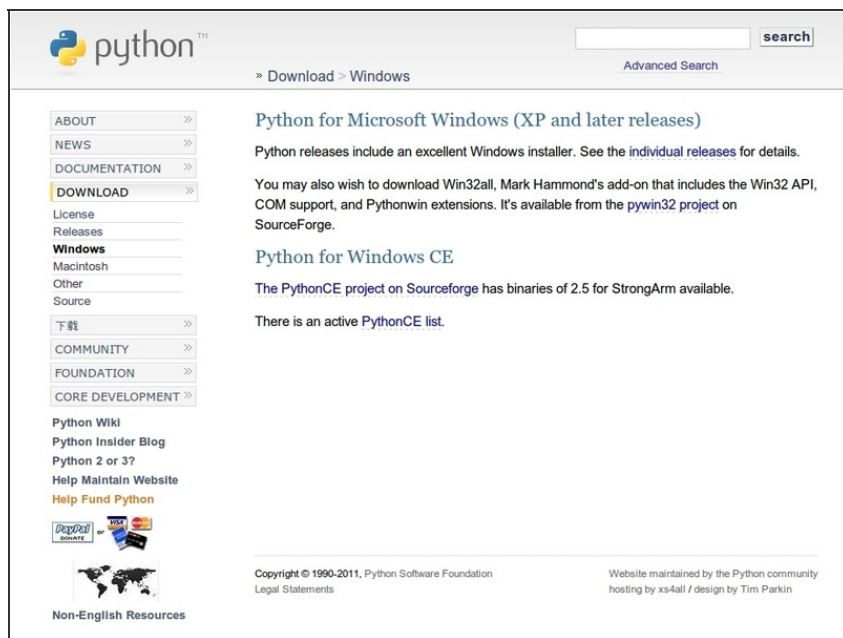
Written By: Chandler

SUMMARY

Programming is one of the key components in the technology in our lives today. From things like video games to even the computer on which you are reading this, programs are needed to make them operate. And if you want to learn how to program, this tutorial is where to start.

In this tutorial, you will learn about the programming language *Python* (yes, that is the name of a snake, but in this case the language was named after the television show *Monty Python*). Python is a great language to start with because of the ease with which you can write programs. Python doesn't require a huge structure of classes and methods to get your program running. However, for those of you who are advanced in programming, Python is optionally object-oriented. This makes it quick and simple to write programs when simplicity is needed, while using the same language you can create advanced programs with multiple classes and methods.

Step 1 — Beginning Programming: Learning Python



- **Setup:**
- Welcome to programming! If you have seen my tutorial on OOP (Object-Oriented Programming), let me tell you that this tutorial will be much easier to follow.
- To start, go to the official [Python programming language website](https://python.org) and click on the Download link. Click on your operating system in the list (Windows, Macintosh, or other) and follow the instructions on the page. When done, open the *Python Interactive Prompt*. To do that, search for "IDLE" and open it. IDLE is an IDE (Integrated Development Environment); in this case it stands for Integrated DeveLopment Environment, spelled IDLE after Eric Idle, one of the actors on *Monty Python*. Once you're at the Python interactive prompt, the fun can begin.
- If you can't find IDLE, search for "Python" and see if you can find something along the lines of "Python (Command Line)".

Step 2

```
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)
[GCC 4.5.2] on linux2
Type "copyright", "credits" or "license()" for more information.
==== No Subprocess ====
>>> print "Hi!"
Hi!
>>> |
```

- **Learning Simple Commands:**
- Once in IDLE or Command Line Python, type:
- `print "Hi!"`
- next to the little `>>>` and hit Enter. It prints (writes to the screen) "Hi!" and another `>>>` appears! That was pretty simple, right?
- Now you know the print command. Just type `print` and write what you want it to print after it in quotation marks! (The quotation marks are very important - without them, Python mistakes `Hi!` for a variable.)

Step 3

- Note: since Python version 3, `print` can be used without parentheses around what you want to print. If you are using a later version, or are experiencing problems trying to use this newer version, just use `print("Text goes here")`.

Step 4

```
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)
[GCC 4.5.2] on linux2
Type "copyright", "credits" or "license()" for more information.
==== No Subprocess ====
>>> print "Hi!"
Hi!
>>> x = 5
>>> print x
5
>>> |
```

- Let's try something new! Variables are just like they were in algebra: they give names to numbers whose values may or may not be known. Python's variables are very similar. I'll show an example.

- `x = 5`
- `print x`
- Did you see what it did? It stored the number 5 into the variable `x`. Then, whenever you say `x` in a program, the computer thinks "5!" So `print x` is equivalent to `print 5`. Make sense so far?

Step 5

```
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)
[GCC 4.5.2] on linux2
Type "copyright", "credits" or "license()" for more information.
==== No Subprocess ====
>>> print "Hi!"
Hi!
>>> x = 5
>>> print x
5
>>> y = 5
>>> z = 6
>>> print y + z
11
>>> |
```

- Python can do math, too. Type in:
- `y = 5`
- `z = 6`
- `print y + z`
- Did you guess what would happen before you typed this in? After you typed all of that in, you probably saw "11" if you typed it correctly. It's just like the last example, but it replaces two variables. When it sees `y`, it replaces it with 5. Same for `z` and 6. So when it comes across `print y + z`, it sees `print 5 + 6`!
- Note: you can also do subtraction (`-`), multiplication, (`*`), division (`/`), and much more.

Step 6

```
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)
[GCC 4.5.2] on linux2
Type "copyright", "credits" or "license()" for more information.
==== No Subprocess ====
>>> print "Hi!"
Hi!
>>> x = 5
>>> print x
5
>>> y = 5
>>> z = 6
>>> print y + z
11
>>> y += 4
>>> print y
9
>>> |
```

- Say you want to add 4 to y. You could type this:
- `y = y + 4`
- It does work. It may seem confusing, but all it's doing is setting y (on the left side) to 4 more than it already was. Rather than doing that, Python has a shortcut that makes things much less confusing:
- `y += 4`
- After typing the code above, print y by typing the following code:
- `print y`
- It will, as you may have guessed, print out "9".

Step 7

```
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)
[GCC 4.5.2] on linux2
Type "copyright", "credits" or "license()" for more information.
==== No Subprocess ====
>>> print "Hi!"
Hi!
>>> x = 5
>>> print x
5
>>> y = 5
>>> z = 6
>>> print y + z
11
>>> y += 4
>>> print y
9
>>> if y == 9:
    print "Y is equal to 9!"

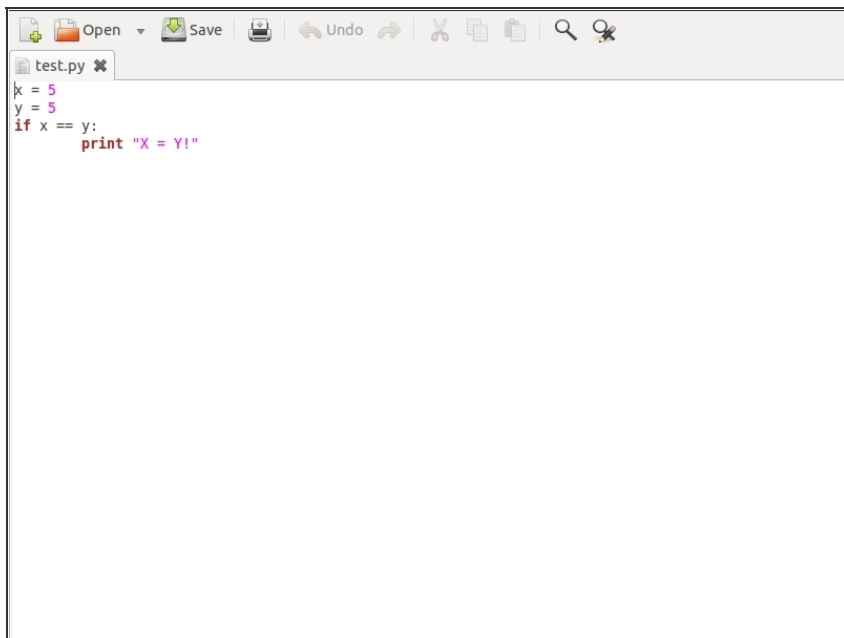
Y is equal to 9!
>>>
```

- Now that you've got that done, let's move on to letting your code make its own decisions. We aren't going to make an AI program (sadly), but we will learn about the `if` statement. Have you ever had a thought that began with "What if..."? For instance, "What if I forgot to close the door?" The answer can vary in intensity, but one example could be that a lot of money might be wasted on your heater. The `if` statement works in a very similar manner.

- Here's how it works:
- `if statement:`
- `[tab] actions`
- Here's an example. (Note: IDLE automatically adds the tab, and for IDLE and command-line Python, in this example, you have to press Enter two times after the last statement.)
- `if y == 9:`
- `[tab] print "Y is equal to nine!"`
- The `"=="` might be confusing, but `"="` was already taken for giving values to variables, so Python had to find a new symbol for `if` statements to indicate equality. Think about it this way: `"="` is like a statement, and `"=="` is like a question (Does y = 9?). There are

also "<" (less than), ">" (greater than), "!=" (not equal to), "<=" (less than or equal to), and ">=" (greater than or equal to).

Step 8



```
test.py
x = 5
y = 5
if x == y:
    print "X = Y!"
```

- Nice job so far. If you've made it here, you will likely make it through the entire series of tutorials. Before I end this tutorial, I'd like to show one last trick: executing Python files.
- Open a new text document, and name it **test.py**. You can also go to File -> New Window in IDLE. Inside, type in the following code:
 - `x = 5`
 - `y = 5`
 - `if x == y:`
 - `[tab] print "X = Y!"`
- Now, go to Terminal, the Command Prompt, or whatever alternative you have on your system and change your directory to where your file is.
- To do that, type `cd` (for all systems) and then the directory that your file is in. For instance (Linux example), type `cd /home/username/Desktop`. Once done with that, type `python` and then the name of your file (also all systems). Example: `python test.py`.